CIS 41A - Lab 5: Regular expression, function as first order object, variable length argument list, generator.

Write an application that lets the user look up the most popular baby names in the US within one decade.

**Input file**

The app uses data from the Social Security Administration (SSA), which records all names given to babies born in the US. The most popular names are displayed on the SSA web pages such as [this](https://www.ssa.gov/oact/babynames/decades/names2000s.html) one, which is for the decade starting in the year 2000. The HTML page for the decades starting in 1980, 1990, and 2000 are in 3 text files called lab5\_1980s.txt, lab5\_1990s.txt, and lab5\_2000s.txt. (In CIS 41B your program will go directly to the website to fetch the data.)

**Overview**

The app lets the user look up names based on:

* popularity ranking (most popular, top 5, etc.)
* full name or partial name matching

Lab 5 consists of 2 files that you will turn in: name.py and lab5.py

name.py serves as a 'back end' to the application: it reads in and stores data from the file, and allows searches for data.

lab5.py serves as a 'front end' to the application: it interacts with the user, calls search methods in name.py, and displays the search result in user-friendly format.

**name.py**

Contains a Name class that has the following methods:

* 1. A constructor that will:
* Accept a filename as input argument.
* Read in each line of text until there's a line with baby names.
* Use regular expression to extract a boy name and girl name from any line that contains 2 names. Here's an example line: <td >Daniel</td> <td>203,720</td> <td >Hannah</td> <td>155,698</td></tr>
* Store the boy name in a list of boys' names, store the girl name in a list of girls' names.
* When all the lines have been read in, print a count of how many boy names and how many girl names have been read in.
  1. A method called searchByInitial that will:
* Accept a gender and a sequence of comma separated initials. (An initial is the first letter in a name.)
* Based on the gender, return all the names that start with each of the initials.
* It's your choice how to return the data, but the user should be able to quickly identify the names of one initial vs. all the other names with other initials.
  1. A method called searchByPopularity that will:
* Accept a limit, which is an integer. A limit of 3 means top 3 names, and a limit of 10 means top 10 names.
* If the limit is 0, it means show all the names, and if the limit is larger than the max number of names, set it to the max number of names.
* Return a generator of names, starting from the most popular names (1 boy name and 1 girl name) and ending with the names at the limit.   
  Note that the code must return a generator so that no memory is used by the method to store the found data.

*[1pt extra credit:*

At the end of name.py, add the following code to test your Name class:

*n = Name()*

*data = n.searchByInitial('M', 'C', 'N', 'K') # return boys' names starting with 3 initials C, N, K*

*## add code to print each initial and corresponding names*

*print()*

*data = n.searchByInitial('M', 'C') # return boys' names starting with initial C*

*## use the same code in the above step to print each initial and corresponding names*

*print()*

*print(list(n.searchByPopularity(20))) # print a list of 20 pairs of names in order*

*print()*

*print(list(n.searchByPopularity(1))) # print a list of the top pair of names*

*Demo your code by Wed Nov 27.*

**lab5.py**

Contains a class called UI (for user interface) that has the following attributes and methods:

1. Set a default filename to one of the 3 input files. (It might be interesting to you to choose the decade of your birth or the birth of someone you know well).
2. A constructor that will:

* Loop to keep prompting the user for a filename or let the user press the Enter key for the default file. The loop stops when there is a valid file that can be read in successfully.
* Use the module name.py to create a Name object that's filled with data from the file.

1. A method that lets the user search by popularity. This method will:

* Loop to keep prompting the user for a limit until you get a valid limit. A valid limit is an integer that's larger than 0 or the Enter key. The Enter key is changed to a limit of 0 for max limit.
* Call a Name method to get the generator for popular names.
* With the generator, loop to print 5 pairs of names at a time. After printing 5 pairs of names, wait for the user to press the Enter key before printing the next 5 pairs of names. Continue until there's no more name to print.
* The pairs of names should be printed in column format.
* Note that you should be using the generator so that no memory is used to store a container of found data.

1. A method that lets the user search by initials. This method will:

* Loop to keep prompting the user for a gender , until you get a valid gender.  
  A valid gender can be 'M' or 'F' or the entire word 'Male" or 'Female', case insensitive, and extra spaces in front or after is okay, but no other character is allowed.  
  The code should only use 1 if statement (no nested ifs) to check for valid input
* Loop to keep prompting the user for a list of comma separated initials, until you get a valid list. A valid list of initials has:
  + one or more initials, and if there are more than one initials, they must be separated by a comma
  + each initial must be a single lower or uppercase letter
  + the comma can have space before or after it

The code should only use 1 if statement (no nested ifs) to check for a valid list.

* Call a Name method to get the names by initials.
* For each initial: print the initial, then print a list of comma separated names. The initials are printed in alphabetical order.

1. A method called run. This method will:

* Loop to print the menu below and ask the user for one of 3 choices, until there is a valid choice.  
   i. Search by initials

p. Search by popularity  
 q. Quit

* When there is a valid choice that's not 'q', process the choice *without having to use an if statement.* Take advantage of Python's features.

At the end of lab5.py, write these 2 lines of code to run the program. This should be the only 2 lines.  
 app = UI()  
 app.run()

**Test**

Use the following sample output for some of the test cases for your code. This does not cover all test cases, so make sure you add your own test cases.

Sample output (user input is in blue)

Enter filename or press Enter key for default file: abc

Error opening file abc

Enter filename or press Enter key for default file: # Enter key

Found 200 names of each gender in file lab5\_2000s.txt

i. Search by initials

p. Search by popularity

q. Quit

Your choice: i

Enter gender: feMAle # extra spaces, case insensitive, whole word

Enter a comma separated list of initials: k # no comma if only one initial

K: Kayla, Kaitlyn, Katherine, Kaylee, Kimberly, Katelyn, Kylie, Katie, Kennedy, Kathryn, Kelsey, Kelly, Kylee, Kendall, Kate, Karen, Kiara, Karina

i. Search by initials

p. Search by popularity

q. Quit

Your choice: i

Enter gender: m # no spaces, case insensitive, one character

Enter a comma separated list of initials: kz,r

Initials must be 1 letter each, separated by comma

Enter a comma separated list of initials: # Enter key

Initials must be 1 letter each, separated by comma

Enter a comma separated list of initials: k , z,r # extra spaces, no space

K: Kevin, Kyle, Kaleb, Kaden, Kenneth

R: Ryan, Robert, Richard, Riley, Ricardo, Raymond # initials are in alphabetical order

Z: Zachary

i. Search by initials

p. Search by popularity

q. Quit

Your choice: p

Enter top number of names or press Enter for all names: 8

press Enter for next 5 names # Enter key

Jacob Emily

Michael Madison

Joshua Emma

Matthew Olivia

Daniel Hannah

press Enter for next 5 names # Enter key

Christopher Abigail

Andrew Isabella

Ethan Samantha

End of list of names

i. Search by initials

p. Search by popularity

q. Quit

Your choice: p

Enter top number of names or press Enter for all names: 2.5

Limit is a positive integer

Enter top number of names or press Enter for all names: 1

press Enter for next 5 names

Jacob Emily

End of list of names

i. Search by initials

p. Search by popularity

q. Quit

Your choice: q